



(19)

Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0 814 583 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.12.1997 Bulletin 1997/52

(51) Int Cl. 6: H04L 12/56

(21) Application number: 97480020.3

(22) Date of filing: 23.05.1997

(84) Designated Contracting States:
DE FR GB

- Brun, Gérard
06100 Nice (FR)
- Galand, Claude
06800 Cagnes Sur Mer (FR)
- Maurel, Olivier
06610 Le Cannet (FR)
- Nicolas, Laurent
06270 Villeneuve Loubet (FR)

(30) Priority: 20.06.1996 EP 96480086

(71) Applicant: International Business Machines Corporation
Armonk, N.Y. 10504 (US)

(72) Inventors:
• Bertin, Olivier
06200 Nice (FR)

(74) Representative: de Pena, Alain
Compagnie IBM France
Département de Propriété Intellectuelle
06610 La Gaude (FR)

(54) **Method and system for minimizing the connection set up time in high speed packet switching networks**

(57) The present invention is directed to a high speed packet switching network and, in particular to a method and system for minimizing the time to establish a connection between an origin and a destination node. Due to high dynamicity of the traffic on transmission links, it is important to select a routing path according to a fully up-to-date information on all network resources. The simpler approach is to calculate a new path for each new connection request. This solution may be very time consuming because there are as many path selection operations as connection set up operations. On another hand, the calculation of paths based on an exhaustive exploration of the network topology, is a complex operation which may also take an inordinate amount of re-

sources in large networks. Many of connections originated from a network node flow to the same destination network node. It is therefore possible to take a serious benefit in reusing the same already calculated paths for several connections towards the same node. The path calculated at the time the connection is requested is recorded in a Routing Database and updated each time a modification occurs in the network. Furthermore, alternate paths for supporting non-disruptive path switch on failure or preemption, and new paths towards potential destination nodes can be calculated and stored when the connection set up process is idle. These last operations are executed in background with a low processing priority and in absence of connection request.

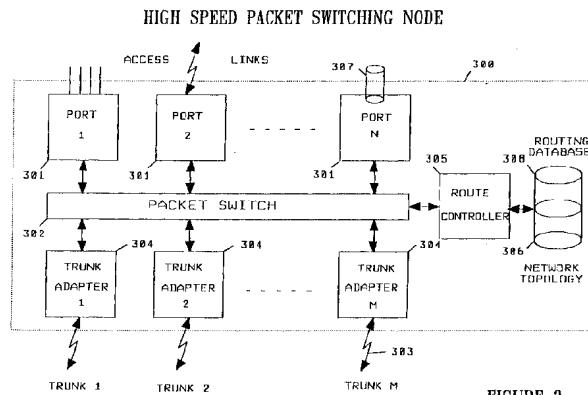


FIGURE 3

Description

Technical Field

The present invention relates to high speed packet switching networks and more particularly to a method and process for minimizing the time to select an optimal routing path between an origin and a destination node in large communication networks.

Background art

High Speed Packet Switching Networks

Data transmission is now evolving with a specific focus on applications and by integrating a fundamental shift in the customer traffic profile. Driven by the growth of workstations, the local area networks interconnection, the distributed processing between workstations and super computers, the new applications and the integration of various and often conflicting structures - hierarchical versus peer to peer, wide versus local area networks, voice versus data - the data profile has become more bandwidth consuming, bursting, non-deterministic and requires more connectivity. Based on the above, there is strong requirement for supporting distributed computing applications across high speed networks that can carry local area network communications, voice, video and traffic among channel attached hosts, business, engineering workstations, terminals, and small to intermediate file servers. This vision of a high speed multi-protocol network is the driver for the emergence of fast packet switching networks architectures in which data, voice, and video information is digitally encoded, chopped into small packets and transmitted through a common set of nodes and links.

An efficient transport of mixed traffic streams on very high speed lines means for these new network architecture a set of requirements in term of performance and resource consumption which can be summarized as follows :

a very high throughput and a very short packet processing time, a very large flexibility to support a wide range of connectivity options, an efficient flow and congestion control.

Throughput and Processing Time

One of the key requirement of high speed packet switching networks is to reduce the end to end delay in order to satisfy real time delivery constraints and to achieve the necessary high nodal throughput for the transport of voice and video. Increases in link speeds have not been matched by proportionate increases in the processing speeds of communication nodes and the fundamental challenge for high speed networks is to minimize the processing time and to take full advantage of the high speed/low error rate technologies, most of

the transport and control functions provided by the new high bandwidth network architectures are performed on an end to end basis. The flow control and particularly the path selection and bandwidth management processes are managed by the access points of the network which reduces both the awareness and the function of the intermediate nodes.

Connectivity

In high speed networks, the nodes must provide a total connectivity. This includes attachment of the user's devices, regardless of vendor or protocol, and the ability to have the end user communicated with any other device. The network must support any type of traffic including data, voice, video, fax, graphic or image. Nodes must be able to take advantage of all common carrier facilities and to be adaptable to a plurality of protocols. All needed conversions must be automatic and transparent to the end user.

Congestion and Flow Control

Communication networks have at their disposal limited resources to ensure an efficient packets transmission. An efficient bandwidth management is essential to take full advantage of a high speed network. While transmission costs per byte continue to drop year after year, transmission costs are likely to continue to represent the major expense of operating future telecommunication networks as the demand for bandwidth increases. Thus considerable efforts have been spent on designing flow and congestion control processes, bandwidth reservation mechanisms, routing algorithms to manage the network bandwidth. An ideal network should be able to transmit an useful traffic directly proportional to the traffic offered to the network and this as far as the maximum transmission capacity is reached. Beyond this limit, the network should operate at its maximum capacity whatever the demand is.

Routing Modes

A general problem in the communication networks is to find a path between a source and a destination node. When the network is using datagrams, the path selection must be done individually for each packet. For virtual circuits, the path decision is done once only at the time of the connection (or session) establishment. In both cases, the choice of a routing algorithm is not easy because it must satisfy a large number of often conflicting requirements. However, the routing algorithm must allow to exploit the network in an optimal way, according to a criterion which can vary with the utilization type. In most of the cases, the network is realized in a way to minimize the packet transit time and to transfer the maximum number of packets. In other cases, the objective is to decrease the communication cost, or to

develop a reliable network able to operate correctly either in case of catastrophic line, node failure or peaks of traffic.

Because of the variety of the constraints, there are a large number of different routing types like flooding routing, random or stochastic routing, deterministic routing. This last routing technique can be implemented according to particular modes such as fixed or adaptive routing, centralized or distributed routing, node by node or end to end routing, connection oriented or connectionless routing....

Adaptive Routing

Contrary to the Fixed Routing, where the routing rules are established once for all, the purpose of the Adaptive Routing is to satisfy at any time the optimization criteria. Tables are permanently updated according for example, the instantaneous state of the traffic on the links.

Distributed Routing

When the characteristics of the network fluctuate, it is possible to adapt the routing by assigning to one node the responsibility to update periodically the routing tables according to the traffic and the topology. The principal disadvantage of this method called Centralized Routing is to generate an important auxiliary traffic and to subordinate the good functioning of the network to only one node. On the other hand, the Centralized Routing can generate some problems at the time the tables are refreshed because said tables cannot be received at the same time by all the nodes. The solution is to decentralize the tables at the level of each node. The Distributed Routing is a method in which neighboring nodes are exchanging messages concerning the traffic and the network conditions to update their own routing table.

End-to-End Routing

The Adaptive and Distributive Routing are both currently used together in most high speed networks. In order to minimize the processing time and to take full advantage of the high speed/low error rate technologies, the transport and control functions provided by the high bandwidth networks are performed on an end to end basis. No hop by hop error recovery or retransmission is envisioned in high speed, high performance (low error) links and thus, there is no need for transit nodes to be aware of individual transport connections. The originating node is responsible for calculating the route that a packet must take through the network. The routing of the packets presents two aspects for which a wide range of implementing methods exists :

1. Determining what the route for a given connection shall be,

2. Actually switching the packet within a switching node.

Connection-Oriented Routing

One distinguishing characteristic of a network is the presence or absence of a "connection" between end users. There are many ways of constructing connectionless or connection-oriented networks. In most of cases, packets using different routing modes can share the same data transmission facilities. Most of the high speed connections are established on a reserved path to guarantee the bandwidth and the quality of service requested by the user. The path across the network is computed in response to each connection request by the originating node. The computed path is based on the parameters characterizing the network connection's requirements and on link capacity and load information maintained within each network node. The originating node sends a reservation request to the end node. As the bandwidth request packet flows along the chosen path, each transit node determines whether it has enough capacity to accept the new connection. If the connection is accepted, the requested bandwidth is reserved. Changes are reflected in every node of the network by means of control messages. Once a connection is established, there is no need to place a destination address in the packet header every time a packet is sent. All that is needed is an identifier to specify which connection is to be used for this packet. Due to the low packet overhead, the connection oriented routing technique is particularly adapted to the transmission of very short packets (for example real-time voice connections). This technique requires that connection tables be set up and maintained dynamically in each node and this restricting makes datagram transport quite inefficient. The implementation of flow and congestion control in a connection oriented network is easier than in a connectionless one because network nodes can regulate the flow on individual connections. However, when a link or a node becomes inoperative (goes down) connections that were passing through the affected link or node are typically lost. A new connection must be established through a different route. This takes time and may disrupt the connection at the end user level. In comparison, connectionless networks typically reroute traffic automatically around link or node failures.

Path Selection

The role of the Path Selection process is to determine optimum paths for users across the network each time a connection is requested. This implies the allocation of network resources to users in order to guarantee their quality-of-service requirements while optimizing the overall throughput within the network. This function takes place entirely within the origin node. Various quality of service parameters may be specified by the users,

some of them in order to satisfy realtime delivery constraints, others related to non real time data traffic transfer... The origin node computes a path to the destination node that is capable of carrying the new connection and providing the level of service required by the new connection. The Path Selection algorithm uses data describing the current traffic load in the entire network (nodes and links). Such data are stored in a topology database located in each node of the network. If no suitable path can be found to meet all requirements, the connection is rejected. Once, the origin node has found a suitable path, a set up message is generated which traverses the selected route, updating the resource allocations for each link visited by the set up message. To meet high throughput, paths are selected and resources reserved once only at the time of the connection establishment. The Path Selection algorithm takes into account various constraints which comes both from the user (quality of service requirements, user's traffic characteristics) and from the current network topology and bandwidth allocation. In addition, the algorithm maximizes the network throughput by choosing a path with the least number of hops and which tends to achieve an even distribution of the traffic among the links. Once an appropriate path has been selected, the network connection establishment process takes place, and only then are the resources along the path reserved.

Connection Set Up Delay

Selecting a path and setting up a connection can take considerable processing overhead in network nodes and can generate a significant delay. For end nodes supporting multiple connections set up requests simultaneously, it is easy to have some sort of scheduling to avoid bottlenecks in the connection set up process. But nevertheless, the time for establishing a connection may be huge. For permanent connections the set up delay probably does not have very much meaning. However, for connections service in real-time, this is a very important parameter, especially since some connections terminate in equipment that runs time-outs. This delay is also important from the point of view of path switching on failure or preemption. The rerouting of a lot of connections that were using a failing link or node may lead to a high number of simultaneous new path set up. The establishment in parallel of multiple new connections through alternate routes take time and usually disrupt the connections at the end user level. The non-disruptive path switch mechanism suppose that :

- first, the physical network has enough residual capacity to accommodate the network connections affected by the failure event, and
- second, the path selection delay is minimized in order to preserve the connections at the end user level.

Large network are highly dynamic and ideally, the Path Selection algorithm should be executed each time a new connection is established to take into account the most recent image of the network. Practically, such a computation would demand an inordinate amount of processing time and is rarely justified during the life of a connection. It would be a waste of resource and money to use a very efficient processor for supporting a burst of path selections at set up time or when a failure occurs and then to keep it idle for hours. On the other hand, it is not useful to select a path based on a perfect image of the network when the status of links and nodes evolves continually. More the demand for bandwidth reservation is available, more the precision required for finding a path can be low.

In large high speed packet switching networks, the support of a high number of connection set up operations implies some requirements in term of performance and resource consumption which can be summarized as follows :

- The origin node - or the node providing the route calculation for the origin node - must be able to decide where to route an incoming packet with a guaranteed quality of service (delay, loss, probability...) in a very short portion of time. The computation must be sufficiently rapid to select an optimum path for each connection request.
- In case of node or link failure, the origin node must be able to establish alternative connections without disrupting the traffic at the end user level.
- The processing time in nodes must be minimized.
- The network resources in the origin node, along the selected path and within the entire network must be optimized.

To optimize the node resources (processing capacity) while minimizing the connection set up delay, routes from the origin node to any destination node in the network can be pre-calculated independently of any connection request. This pre-calculation is executed in background when the connection set up process is idle. When a connection is requested, the path selection process first searches for a pre-calculated route. That is only when no pre-calculated path can be found, that the path selection algorithm is triggered. The path is computed and used for establishing the connection and routing the packets within the network. Said path is also stored in a list of pre-selected paths.

In order to optimize the network resources in term of overall throughput, the Path Selection process takes into account the current traffic conditions on the links. Consequently, the pre-calculated paths must be continuously updated according to the load variation on the network.

In case of link failure, it is important to reroute the traffic on alternative paths without any disruption of the end user connections. In general, the closer to the des-

mination node the link is, the more the link carries connections towards this destination node. A failure on a link adjacent to an origin or destination node may entail a disruption of a lot of connections and may invalidate several pre selected routes. A solution is to pre-calculate more than one route for every destination node, each route using different links adjacent to the origin and destination nodes.

Therefore, a method must be provided for determining as fast as possible a route which fulfills the connection requirements in term of quality of service while optimizing the network resources.

Summary of the Invention

The present invention is directed to packet switching communication networks comprising a plurality of nodes interconnected with transmission links. The object of the present invention is to minimize in access nodes the connection set up delay, and in particular the time to select an optimal path throughout the network between the access node and a destination node. Each node in the network comprises one or more communication adapters for receiving and transmitting data packets, a Route Controller for allocating, controlling, and managing network resources, a Topology Database for storing the network configuration and traffic characteristics, updated by means of network control messages, and a Routing Database, updated simultaneously with the Topology Database, for storing the selected or computed paths with their characteristics.

For each connection request, the claimed method and system involves the steps of :

- * for each connection request, selecting a pre-calculated path satisfying said connection request in the routing database,
- * if no pre-calculated path satisfying the connection request is already stored in the routing database, calculating a path satisfying said connection request and storing said path within the routing database.

In absence of connection request, the method and system involves the further steps of :

- * re-calculating the paths stored in the routing database periodically or in response to pre-determined events such as traffic variations, network configuration changes...
- * calculating, for each pre-calculated path stored in the routing database, at least one alternate path with the same origin node and the same destination node.
- * calculating paths from the origin node to all possible destination nodes.
- * removing pre-calculated paths from the routing database after a predetermined period of time without

being selected.

Brief Description of the Drawings

- 5 Figure 1 describes the call set up process according to the present invention.
- 10 Figure 2 shows a typical model of high speed packet switching network including the access and transit nodes claimed in the present invention.
- 15 Figure 3 describes a high speed Routing Point according to the present invention.
- 20 Figure 4 shows the Routing Database structure according to the present invention.
- 25 Figure 5 describes the Topology Database structure.
- 30 Figure 6 shows the link characteristics stored in the Topology Database.
- 35 Figure 7 shows a detailed flow chart of the Search Path Procedure according to the present invention.
- 40 Figure 8 shows a detailed flow chart of the Store Path Procedure according to the present invention.
- 45 Figure 9 shows a detailed flow chart of the Alternate Path Computation Procedure according to the present invention.
- 50 Figure 10 shows a detailed flow chart of the Potential Path Computation Procedure according to the present invention.
- 55 Figure 11 shows a detailed flow chart of the Path Recomputation Procedure according to the present invention.
- 60 Figure 12 shows a detailed flow chart of the Link Update Procedure according to the present invention.
- 65 Figure 13 shows a detailed flow chart of the Garbage Collection Procedure according to the present invention.
- 70 Description of the preferred embodiment
- 75 **High Speed Communications**
- 80 As illustrated in Figure 2, a typical model of communication system is made of several user networks (212) communicating through a high performance network (200) using private lines, carrier provided services, or public data networks. Each user network can be described as a set of communication processors and links (211) interconnecting large computers used as enterprise servers (213), user groups using workstations or personal computers attached on LAN (Local Area Networks 214), applications servers (215), PBX (Private Branch eXchange 216) or video servers (217). These user networks, dispersed in different establishments, need to be interconnected through wide area transport facilities and different approaches can be used for organizing the data transfer. Some architectures involve the checking for data integrity at each network node, thus slowing down the transmission. Others are essentially looking for a high speed data transfer. To that end

the transmission, routing and switching techniques within the nodes are optimized to process the flowing packets toward their final destination at the highest possible rate.

The present invention belongs essentially to the latter category and more particularly to the fast packet switching network architecture detailed in the following paragraphs.

High Performance Packet Switching Networks

The general view in Figure 2 shows a fast packet switching transmission system comprising eight nodes (201 to 208) each node being interconnected by means of high speed communication lines called Trunks (209). The access (210) to the high speed network by the users is realized through Access Nodes (202 to 205) located at the periphery. These Access Nodes comprise one or more Ports, each one providing an access point for attaching external devices supporting standard interfaces to the network and performing the conversions required to transport the users data flow across the network from and to other external devices. As example, the Access Node (202) interfaces respectively a Private Branch eXchange (PBX), an application server and a hub through three Ports and communicates through the network by means of the adjacent Transit Nodes (201), (205) and (208).

Switching nodes

Each network node (201 to 208) includes a Routing Point where the incoming data packets are selectively routed on the outgoing Trunks towards the neighboring Transit Nodes. Such routing decisions are made according to the information contained in the header of the data packets. In addition to the basic packet routing function, the network nodes provide ancillary services such as:

- * the determination of routing paths for packets originated in the node,
- * directory services like retrieving and updating information about network users and resources,
- * the maintaining of a consistent view of the physical network topology, including link utilization information, and
- * the reservation of resources at access points of the network.

According to the present invention, these ancillary services include:

- * the storage of routing paths within the node
- * the updating of these paths.

Each Port is connected to a plurality of user processing equipments, each user equipment comprising either a source of digital data to be transmitted to

another user system, or a data sink for consuming digital data received from another user system, or, typically, both. The interpretation of the users protocols, the translation of the users data into packets formatted appropriately for their transmission on the packet network (200) and the generation of a header to route these packets are executed by an Access Agent running in the Port. This header is made of Control, Routing and Redundancy Check Fields.

- 10 * The Routing Fields contain all the information necessary to route the packet through the network (200) to the destination node to which it is addressed.
- 15 * These fields can take several formats depending on the routing mode specified (connection oriented or connectionless routing mode...).
- 20 * The Control Fields include, among other things, an encoded identification of the protocol to be used in interpreting the Routing Fields.
- * The Redundancy Check Fields are used to check for errors in the header itself. If an error is detected, the packet is discarded.

Routing Points

Figure 3 shows a general block diagram of a typical Routing Point (300) such as it can be found in the network nodes (201 to 208) illustrated in Figure 2. A Routing Point comprises a high speed packet Switch (302) onto which packets arriving at the Routing Point are entered. Such packets are received:

- 35 * from other nodes over high speed transmission links (303) via Trunk Adapters (304).
- * from users via application adapters called Ports (301).

Using information in the packet header, the adapters (304, 301) determine which packets are to be routed by means of the Switch (302) towards a local user network (307) or towards a transmission link (303) leaving the node. The adapters (301 and 304) include queuing circuits for queuing packets prior to or subsequent to their launch on the Switch (302).

The Route Controller (305) calculates the optimum paths through the network (200) so as to satisfy a given set of quality of services specified by the user and to minimize the amount of network resources used to complete the communication path. Then, it builds the header of the packets generated in the Routing Point. The optimization criterion includes the number of intermediates nodes, the characteristics of the connection request, the capabilities and the utilisation of the links (Trunks) in the path, the number of intermediate nodes... According to the present invention, the optimum route is stored in a Routing Database (308) for further reuse.

All the information necessary for the routing, about

the nodes and transmission links connected to the nodes, are contained in a Network Topology Database (306). Under steady state condition, every Routing Point has the same view of the network. The network topology information is updated when new links are activated, new nodes added to the network, when links or nodes are dropped or when link loads change significantly. Such information is exchanged by means of control messages with all other Route Controllers to provide the up-to-date topological information needed for path selection (such database updates are carried on packets very similar to the data packets exchanged between end users of the network). The fact that the network topology is kept current in every node through continuous updates allows dynamic network reconfigurations without disrupting end users logical connections (sessions).

The incoming transmission links to the packet Routing Point may comprise links from external devices in the local user networks (210) or links (Trunks) from adjacent network nodes (209). In any case, the Routing Point operates in the same manner to receive each data packet and forward it on to another Routing Point as dictated by the information in the packet header. The fast packet switching network operates to enable a communication between any two end user applications without dedicating any transmission or node facilities to that communication path except for the duration of a single packet. In this way, the utilisation of the communication facilities of the packet network is optimized to carry significantly more traffic than would be possible with dedicated transmission links for each communication path.

Network Management

Network Control Functions

The Network Control Functions are those that control, allocate, and manage the resources of the physical network. Each Routing Point has a set of the foregoing functions in the Route Controller (305) and uses it to facilitate the establishment and the maintenance of the connections between users applications. The Network Control Functions include in particular:

* **Directory Services**

- for retrieving and maintaining information about network users and resources.

* **Bandwidth Management**

- for processing the bandwidth reservation and maintenance messages, and
- for monitoring the current reservation levels on links.

* **Path Selection**

- for choosing the best path for each new connection considering the connection requirements and the current link utilisation levels.

5 * **Control Spanning Tree**

- for establishing and maintaining a routing tree among the network nodes,
- for using it to distribute control information (in parallel) including link utilisation, and
- for updating the Topology Database of the nodes with new network configurations or link/node failures.

15 * **Topology Update**

- for distributing and maintaining, using the Spanning Tree, information about the logical and physical network (including link utilization information) in every node.

* **Congestion Control**

- for enforcing the bandwidth reservation agreements between the network's users and the network which are established at the call set up time, and
- for estimating actual bandwidth and for adjusting reservation if necessary during the life of the connection.

Topology Database (TDB)

The Topology Database contains information about nodes, links, their properties, and the bandwidth allocation. The topology information is replicated in each node of the network. An algorithm guarantees the correctness of each node's Topology Database when links and nodes are added or deleted or when their characteristics change. The database comprises:

- * the physical topology of the network which includes static information like physical characteristics of nodes and links,

- 45 * the state of nodes and links, and
- * the link utilisation which includes dynamic characteristics like current bandwidth (used and reserved), real time measurements...

50 The general organisation of the Topology Database is shown in Figure 5. To each resource in the network, nodes (501) or links (502), is associated an entry in the database. In particular, each link entry includes the following characteristics:

- 55 * (503) the **link physical properties**:

- transmission medium and speed,

<ul style="list-style-type: none"> - routing mode supported, - maximum packet size, - link buffer capacity, - propagation delay, - bandwidth reservation supported... <p>* (504) the link state:</p> <ul style="list-style-type: none"> - on-line (link can accept user connections), - quiesce (link cannot accept additional user connections, but existing connections continue), - off-line (link cannot accept user connections and existing connections are cancelled)... <p>* (505) the link utilization:</p> <ul style="list-style-type: none"> - real time measurements, - reserved bandwidth, ... 	5 10 15	<p>by the user. However, for bursty user traffic, this approach can waste a significant amount of bandwidth across the network. To save resources, the bandwidth amount actually reserved is equal to an "Equivalent Capacity" $\hat{C}_{k,ij}$ said Equivalent Capacity being a function of the source characteristics and of the network status. The bandwidth reservation falls somewhere between the average bandwidth required by the user and the maximum capacity of the connection.</p> <p>The value</p> $\hat{C}_{R,ij} = \hat{C}_{R,ij}^K = \sum_{k=1}^K \hat{C}_{k,ij} = \text{sum}$
---	---------------	---

Figure 6 shows in a table, some of the information stored in the Topology Database. Though all characteristics of the links are listed in each node, in the present application only a few will be described:

* **Total Capacity (bps)** C_{ij}

The topology Database contains, for each link, its Total Capacity. The value C_{ij} represents the total bandwidth available on the link between nodes i and j .

* **Reservable Fraction (%) rf**

As might be expected, one of the critical characteristics of transmission links is the fraction of the link capacity effectively available. Links cannot be loaded up to a theoretical maximum load (bandwidth) for two reasons:

- first, to set aside bandwidth for network control functions, and
- secondly, to keep the loss probabilities and queuing delays low in the case of short term bandwidth violations by the different traffic sources.

The reservable fraction of a link rf is the effective percentage of the Total Capacity C_{ij} that can be reserved on the link between nodes i and j to maintain a reasonable quality of transmission. If C_{ij} is the Total Capacity of the link, then $R_{ij}=rf \times C_{ij}$ is the Reservable Capacity of this link ($\hat{C}_{ij} \leq R_{ij} \leq C_{ij}$).

Note : For most network architectures, no more than 85% of the total bandwidth of a link C_{ij} can be explicitly reserved for user traffic ($rf < 0.85$).

* **Total Reserved Equivalent Capacity (bps)** $\hat{C}_{R,ij}$

For a connection k on a link between nodes i and j , the simplest way to provide low/no packet loss would be to reserve the entire bandwidth requested

20 of the reserved Equivalent Capacities represents the total bandwidth reserved on the link between nodes i and j by K connections already established. If the difference between this already link Equivalent Capacity $\hat{C}_{R,ij}$ and the Total Reservable Capacity of the rf C_{ij} is less than 25 the bandwidth requested by a new reserved connection then the link cannot be selected. However, the link may be selected for a non-reserved connection where no explicit bandwidth reservation is needed.

Note : the Equivalent Capacity is calculated from 30 values called **"link metrics"** representing the aggregation of all connections (possibly with varied characteristics) traversing a link. Link metrics are distributed to other nodes via a topology update broadcast.

35 * **Total Bandwidth used by Non-Reserved Traffic (bps)** $M_{NR,ij}$ The value $M_{NR,ij}$ represents the total load or bandwidth currently used by non-reserved traffic as measured on the link between nodes i and j .

40 * **Maximum Packet Size (bytes)** mps_{ij} mps_{ij} is defined as the maximum packet size supported by the link between nodes i and j .

45 * **Total Bandwidth Used (bps)** $\hat{C}_{T,ij}$ The Total Bandwidth Used $\hat{C}_{T,ij}$ on the link between node i and node j is computed by adding the total reserved bandwidth $\hat{C}_{R,ij}$ and the measured bandwidth $M_{NR,ij}$ used by non-reserved traffic.

50 **Bandwidth Management**

55 Users are requiring different quality-of-services. In order to provide the various service levels, different types of network connections are established. A connection is defined as a path in the network between the origin access node and the destination access node representing respectively the source user and the target us-

er. Networks connections can be classified as reserved or non-reserved. Reserved network connections require bandwidth to be allocated in advance along the chosen path.

Most of the high speed connections are established on a reserved path to guarantee the quality of service and the bandwidth requested by the user. This path across the network is computed by the origin node using information in its Topology Database including current link utilizations. The origin node then sends a reservation request along the chosen path, and intermediate nodes (if allowing the reservation) then add this additionally reserved capacity to their total.

These changes are reflected in topology broadcast updates sent by the intermediate nodes. Intermediate nodes need not to have an awareness of the status of each connection on their adjacent links. If an intermediate node does get too many packets, generally because of unanticipated burstiness, it simply discards them (the user can select a service that will recover from such discards).

Depending on the node type, the function of the Bandwidth Management is :

* in the **origin node**,

- to identify the best possible route according to the network status and the connection parameters including the connection priority.
- to reserve at connection setup, the bandwidth required by the network connections and to maintain this bandwidth for the duration of the connection.
- to reject the connection if resources needed to satisfy the request are not available in the network.

in a **transit node**,

- to administer the bandwidth reservations on the links.

Connection Set Up

The connection set up and bandwidth reservation process, as shown in Figure 1, comprises the following steps :

- * (101) a Connection Request is specified by the user via a set of parameters including origin and destination network address, and data flow characteristics (bit rate, burstiness).
- * (102) a **Path Selection** process determines a path and a set of connection requests, one for each link of the path, using parameters provided by the To-

pology Database.

- * (103) a **Bandwidth Reservation** process uses the connection requests to reserve bandwidth on each of the links of the path. This process involves exchange of information (109) between the origin (access) node (100), the transit nodes (107) on the path, and the destination node (108).
- 10 * (104) **Bandwidth Reservation replies** from transit nodes and end node generate either a call acceptance or a call reject (110).
- 15 * (105) a **Link metric Update** process updates, in case of call acceptance, the modified link metrics. This information (111) is sent through the Control Spanning Tree to the Topology Database of each node in the network by means of a broadcast algorithm.
- 20 * (106) a **Congestion Control Set Up** adjusts, if the call is accepted, the network connection characteristics.

Path Selection

The purpose of the Path Selection process is to determine the best way to allocate network resources to connections both to guarantee that user quality of service requirements are satisfied and also to optimize the overall throughput of the network. The Path Selection process must supply to the requesting user a path over the network over which a point-to-point connection will be established, and some bandwidth will be reserved if needed. The Path Selection algorithm uses as input parameters in one hand the user requirements and on the other hand the status of the network links and nodes as maintained in the Topology Database.

Selection Criteria : The Path Selection process takes place entirely within the node wherein the connection is requested. It makes use of the Topology Database and selects the "best path" based on each of the following criteria in order of importance :

* Quality-of-Service :

The connection's quality-of-service requirements are to be satisfied throughout the life of the connection. There are a large number of variables that determine the performance of a network. However, the quality-of-service can be defined as the set of measurable quantities that describe the user's perception of the service offered by the network. Some of the quality-of service parameters are listed below :

- connection set up delay,
- connection blocking probability,
- loss probability,

- error probability,
- end-to-end transit delay,
- end-to-end delay variation,
- ...

Some of these quantities have an effect upon how paths are computed, for example the packet loss probability or the end-to-end transit delay : the sum of propagation delays along a computed path may not violate the end-to-end transit delay specifications.

* Minimum Hop :

The path shall consist of as few links as feasible to support the connection's quality of service requirements, thus minimizing the amount of network resources as well as processing costs to support the connection. The path computation is based on the links utilization at the time the connection is requested.

* Load Balancing :

Among a minimum hop path, a path with "lightly loaded" links is preferred over a path with "more heavily loaded" links based on the network conditions at the time of path selection. The load of a link depend of the customer criteria : it can be an increasing function of the total reserved bandwidth of the link, proportional to the amount of traffic actually measured on the link, ... When the path load (sum of the load of the links over the selected path) is the preponderant criterion of selection, the path of lesser load is chosen.

Satisfying the first requirement is the key factor in path selection and the other two functions are used to optimize traffic through the network.

Path Calculation : The Path Selection process collects all the input information and calculates first the Equivalent Capacity of the new network connection. Then potential paths through the network from the origin to the destination node(s) are calculated. The Topology Database is used to determine the suitability of links for the potential path. The algorithm constructs new potential paths, adding links and nodes on a hop-by-hop basis, checking at each stage that the user requirements are being met.

At each stage, as links are added into a potential path, the algorithm determines that the new link has enough bandwidth to meet the resource reservation requirements. The algorithm also checks, at each stage, that addition of the new link still allow the total path to meet the user quality-of-service requirements. The algorithm is different for the different traffic classes. For example the end-to-end delay variation (jitter) requirements is more important for the real-time graphic calculation than for non-real time.

Additional Requirements : If the Path Selection process finds a number of suitable potential paths for a new network connection, which all meet the quality of

service requirements, then it will select the path with the minimum number of hops as the optimal paths. This is because the minimum-hop path will result in the least amount of reserved bandwidth in the network, and therefore preserve network resources and reduce cost.

- 5 If the Path Selection process detects a number of suitable minimum-hop paths for a new network connection, which all meet the quality-of-service requirements, then it will select a path with more "lightly loaded" links
10 in preference with heavily loaded links. The link utilizations from the Topology Database are used in this calculation.

Routing Database Structure

15 The general organization of the Routing Database (400) is shown in Figure 4. The Routing Database is divided in two tables :

- 20 * The Path Table (410) for recording the routing paths, and
* The Link Table (420) for recording the links related to said routing paths stored in said path Table (410).
25 In a preferred embodiment, Path and Link Tables are implemented as hash tables with direct chaining in an overflow area as described in publication entitled "Algorithms + Data Structures = Programs" for Niklaus Wirth (pages 264 to 274, Prentice-Hall Inc).
30

Path Table

For the present application, only some of the parameters stored in the Path Table (410) will be described. Each entry of the Path Table represents a path between an origin node and a destination node in the network satisfying specific quality of service and traffic requirements :

- 40 * **Destination Node (412)**
Identification of the last node on the path.
* **Path Characteristics (413)**
45 - **Maximum Packet Size** : minimum of the maximum packet sizes supported by all links on the path calculated for each of the different delay priorities (real-time, non-real-time, non reserved delay priorities).
50 - **Maximum delay** : sum of the transfer delay of each link on the path calculated for each of the different delay priorities (circuit emulation real-time, non-realtime, delay priorities).
55 - **Maximum delay variation (Jitter)** : sum of maximum delay variation of each links on the path calculated for each of the different delay

- priorities (real-time, non-real-time, delay priorities). 5
- **Loss probability**: sum of loss probabilities of all links in the path calculated for each of the different delay priorities (real-time, non-real-time, best effort delay priorities). The loss probability may be estimated by the ratio (lost information)/(total information transferred). The lost information is information sent by the origin node that never arrives at the intended destination due to network conditions (congestion, traffic policing, errors...). 10
 - **Minimization criteria**: depending on its type, a connection may pose its quality-of-service requirements and in particular its delay requirements in term of : 15
 - . *End-to-end delay without limit* : among all possible paths, the path with the minimum end-to-end delay is selected. 20
 - . *Maximum absolute end-to-end delay* : the path chosen guarantees that the end-to-end delay will be below the specified value throughout the life of the connection. This type of link delay constant is used mostly for real time traffic, which has a stringent end-to-end delay requirement. 25
 - . *Maximum end-to-end delay variation* (maximum jitter) : the amount by which the delay associated with a service varies is also known as "delay jitter". For some types of services a considerable delay may be acceptable provided that it is predictable, that is, the delay jitter is kept small. For other services (for example, voice services) both the delay and the delay jitter need to be kept small. 30
 - . *Maximum end-to-end delay with playout buffer*; for application involving isochronous traffic (for example, voice), a playout buffer is required at the destination node to smooth out the effects of delay variation and to ensure that packets are sent to the user at regular intervals. The amount of playout delay is selected to equal the maximum end-to-end delay variation that should be cancelled (jitter). The playout delay contributes to the end-to-end delay. 40
 - **Security level (encryption) support**: many organizations are concerned that their internal communications may be intercepted by eavesdroppers. To minimize the likelihood that communications can be intercepted, there are number of measures that can be taken : for example, using communication links that are 45
- harder to intercept such as optical fibers, or using encryption devices. However such security measures may not be required for all services as their use may involve additional expense or inconvenience. Therefore, there is a requirement when establishing a service to specify the security requirements for the service. The specification of the security needs for a service are generally specified using a security level. 55
- **Hop count**: number of links in the path. The path selection process calculates a path with as few links as possible that supports the quality-of-service requirements of each requested connection. However, it may be desirable to put an upper limit on the number of hops in the chosen path. What this means is that there is a certain maximum number of hop which the cost of using the path becomes unacceptably high. In a preferred embodiment, the path selection algorithms check the candidate paths in the increasing number of hops and stop when the imposed maximum number of hop is reached. If more than one minimum-hop path satisfying the quality-of-service requirements is found, the one with the "lightly loaded" links is preferred over a path with "more heavily loaded" links based on the network conditions at the time of path selection. In most of cases, this condition tends to maximize the overall throughput of the network. 60
 - **Bandwidth reservation support**. network connections are primarily classified, based on their network resource needs, into either : 65
 - . *Reserved network connections*. Reserved network connections require the reservation of some network resources prior to successful establishment of the network connection. If the network resources required for a network connection cannot be allocated, establishment of the network connection will fail. Of particular interest are reserved network connections that reserve bandwidth as these types of network connections are capable of providing a quality-of-service. 70
 - . *Non-reserved network connections*. The non-bandwidth-reserved traffic is delayable, has a low average bandwidth, may last a very short time and its burstiness characteristics may not be known. The lowest delay is assigned to non-reserved traffic and the network drops non-reserved packets when their buffer overflow at intermediate nodes. That is why the transfer delay 75

and loss probability are not part of the quality-of-service guaranteed to non-reserved connections.

While the connectionless routing mode is quite efficient to route packets that does not require any reservation nor quality-of-service, connection oriented routing mode is well adapted to route packets when reservation and quality-of-service are expected and provided.

- **Real-time traffic support**: Each type of network traffic has its own particular characteristics and therefore needs to be treated differently. The traffic types may be summarized as follows :

- . traffic that is delay-and loss-sensitive, for example interactive video.
- . traffic that is delay-sensitive but tolerant to moderate losses, for example voice.
- . traffic that is both delay-and loss tolerant, for example most current datagram services.
- . traffic that is delay-tolerant but loss-sensitive, for example most database updates.

In addition to packet loss and average delay, real-time applications (like voice) are sensitive to the variation in delay (jitter) of the packet stream between origin and destination. Meeting the delay loss and jitter requirements of different traffic (or applications) in an integrated network requires corresponding functions to be defined in the architecture so that, when multiplexed together, one type of traffics will not cause degradation to the service provided to another type.

- **Circuit emulation support**: Circuit emulation is intended for constant-rate voice and video applications etc... It provides transport for serial bit streams between access link interfaces on nodes and is used to support services that are similar to the dedicated leased line style of services available from most telecommunication carriers. Circuit emulation requires :

- . a constant bit rate at origin and destination,
- . a timing relationship between origin and destination,
- . a connection between end users of the service.

- **ATM network connection support**: Current high speed packet switching network architecture can support different logical link types. However the choice of link types for connections between nodes in a network depends on the services available from carriers, the link speed required ...

- . Asynchronous Transfer Mode links (ATM) can be dedicated leased lines or ATM connections

provided by an ATM bearer service. Data is transferred on ATM links in fixed-length (53 bytes) cells with a 5 bytes ATM header and a 48 bytes payload. ATM links are used when nodes are connected via a permanent virtual path connection provided by an ATM bearer service.

. Variable-length packet links can be a variety of leased or switched link connections (like T1, T3, E1,) supporting transmission of variable-length packets. Variable-length packet links should be used when the majority of traffic in the network is variable-length data because variable-length packet links require significantly less overhead when transmitting variable-length packet traffic (for example, Frame Relay).

. In some architectures, nodes support any combination of ATM and variable-length packet links to adjacent nodes and user data might be transferred through the network over a sequence of ATM and variable-length packet links without restriction. This combination of links can be used to build a private ATM network, that is, provide an ATM bearer service to ATM attached devices.

30 - **Routing support**: Data packets are routed and queued in the transit nodes according to the routing information contained in the header. Several routing modes can be used in high speed networks. In most of the cases, packets using different modes can share the same data transmission facilities.

35 . **Automatic Network Routing (ANR)** is a particular implementation of the distributed routing for connectionless networks. The access node is responsible for calculating the route the packet must take through the network. Each packet includes in its routing field a list of the labels of the links through which the packet will pass as it moves across the network. ANR requires no connection set up activity in intermediate nodes and is suitable for true datagram services. Variable-length packets and optional features such as a copy function and reverse path accumulation are supported.

40 . **Label Swapping** is used in connection oriented networks. Each packet sent on the link has a header which includes an arbitrary number identifying which logical connection that this packet belongs to. Label Swapping requires that the connection tables be set up and maintained dynamically in each node. In contrast with ATM, Label Swapping supports variable-

length packets, optional features such as reverse path accumulation and a more sophisticated loss priority structure.

- **Asynchronous Transfer Mode (ATM)** is similar to the Label Swapping transfer mode except some features, including the header structure, the support of fixed-size packets (48 bytes payload with a 5 bytes header), the use of a two-level virtual connection hierarchy (Virtual Channel Connection VCC and Virtual Path Connection VPC), ...
- **Free Bandwidth**: minimum of the available bandwidth of all links along the path. Free Bandwidth available on link l_{ij} between nodes i and j : $C_{ij,rf} - \hat{C}_{R,ij}$

* **First link (414)**

Identification of the first link on the path (for example the address of the first link in the Link Table).

* **Time-to-Live (415)**

This parameter is used to rid the Routing Database of outdated information. The "Time-to-Live" counter of each path in the Routing Database is decremented once every day. When a counter reaches zero, the corresponding path is considered as outdated and is removed from the Routing Database. Each time a path is selected, its counter is reset to its maximal value. See Figure 13 for further details concerning the Garbage Collection Procedure.

Link Table

For the present application, only some of the parameters stored in the Link Table (420) will be described. Each entry of the Link Table represents a link l_{ij} between a node i and a node j .

* **Link Identifier in the Topology Database (422):**

The Link Identifier (for example the address of link in the TDB) is a means for having a direct access to the link characteristics (physical properties, state, utilization) stored in the Topology Database (502).

* **Link Characteristics (423):**

- Allocated Bandwidth $\hat{C}_{R,ij}$ stored at different holding priority levels for real-time and non-real time delay priority (no allocated bandwidth in non reserved delay priority).

Note: The holding priority is the preemption priority of a connection once it has been established. On an other hand, the requesting priority defines a network con-

nexion's ability to preempt other connections. An existing connection defends itself against preemption using its holding priority.

- 5 - Equivalent Capacity $\hat{C}_{k,ij}$ of the requesting connection k computed with real-time and non real-time delay priority.

* **Parent Path (424)**

10 Identification of the path associated with the link and stored in the Path Table (for example the address in the Path Table of the Parent Path).

15 * **Next Link in Path (425)**

Identification of the next link along the path if any (for example the address in the Link Table of the next link on the path).

Routing Database Procedures

Garbage Collection

25 The Garbage Collection Procedure is used to remove from the Routing database the outdated information. In a preferred embodiment, this procedure is executed periodically (for example every day). Figure 13 shows a general flow chart of the "Garbage Collection" 30 Procedure.

- * (1300): start of the procedure.
- * (1301): the first path stored in the Routing Database is extracted.

35 * (1302): the "time-to-live" counter associated with this path is decremented.

- * (1303) the value of the "time-to-live" counter is tested for determining whether or not the path is outdated.

- 40
- (1304): if the value of "time-to-live" counter is equal to zero, the path is removed from the Routing Database and the procedure goes on with step (1305).
 - If the value of the "time-to-live" counter is not equal to zero, the procedure goes on with step (1305).

45

- * (1305): a test determines whether or not other paths are stored in the Routing Database.

- 50
- If no other paths are stored, the procedure is complete and terminates in step (1306).
 - (1307): if other paths are stored, the next path stored in the Routing Database is extracted and the procedure goes on with step (1302).

Search Path

Figure 7 shows a general flow chart of the "Search Path" Procedure. This procedure is called during the connection set up. When a connection request is received by the Path Selection, this procedure scans the Routing Database to determine whether or not there is at least one path satisfying this request. Let Z be the destination node and SP the list of selected paths.

- * (700): start of the procedure.
 - * (701): a search in the Path Table determines whether or not a path for the requested destination node Z is already stored.
 - If no path is already stored in the Path Table, the path must be calculated. The Search Path Procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (703): if at least one path is already stored, the first path p stored is extracted from the Path Table.
 - * (704): a test determines whether or not the characteristics (413) of the path p stored in the Path Table satisfy the request.
 - If the characteristics (413) of the path don't satisfy the request, the procedure goes on with step (705).
 - (706): if the characteristics of the path satisfy the request, the first link l_{ij} of the path p is extracted from the Link table.
 - * (707): a test determines whether or not the link between nodes i and j can support the new connection k in terms of:
 - . Bandwidth requested by the new connection : $C_{k,ij}$
 - . Allocated bandwidth on the link: $\hat{C}_{R,ij}$
 - If the link cannot support the new connection k, the procedure goes on with step (705).
 - If the link can support the new connection k, the procedure goes on with step (708).
 - * (708): a test determines whether or not there are more links in path p to scan.
 - (709): if there are more links to scan, the next link l_{ij} in path p is extracted from Link table and the procedure goes on with step (707).
 - (710): if there are no more links to scan, path p
- 5 * (705): a test determines whether or not more paths are stored with the requested destination node Z.
 - (711): if more paths are stored, the next path p is extracted from the Path Table and the procedure goes on with step (704).
 - If no more paths are stored, the procedure goes on with step (712).
- 10 * (712): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (713): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 15 * (714): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (715): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 20 * (716): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (717): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 25 * (718): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (719): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 30 * (720): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (721): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 35 * (722): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (723): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 40 * (724): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (725): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 45 * (726): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (727): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 50 * (728): a test determines whether or not the list of selected paths SP is empty.
 - If the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.
 - (729): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
- 55 * (730): a test determines whether or not the list of selected paths SP is empty.
 - (731): if the list is not empty, the shortest path (with the smallest Length) in SP is selected. If several paths with the same length type exist in SP, the path having the largest free bandwidth is chosen in order to perform load balancing on the network. Then the process is complete and terminates in stop step (714).
 - (732): if the list is empty, no path is available. The procedure is then complete and terminates in stop step (702). The Path Selection process will then determine a path which will be stored in the Routing Database according to the Store Path Procedure described in Figure 8.

Store Path

Figure 8 shows a general flow chart of the "Store Path" procedure. This procedure is called at the end of each path computation processed by the Path Selection.

- * (800): start of the procedure.
- * (801): a test determines whether or not the calculated path is already stored in the Routing Database.
 - If the path is already stored, the procedure is complete and terminates in stop step (802).
 - If the path is not already stored, the procedure goes on with step (803).
- * (803): a test determines whether or not there is a free entry in the memory to store a new path in the Routing Database.
 - (804): if there is no free entry, the path with the lowest free bandwidth is removed from the Routing Database and the procedure goes on with step (805).
 - If there is a free entry, the procedure goes on with step (805).

- * (805): the related links are stored in the Link table (420). Link entries are chained together with the address of the next link on the path (425). The path is linked to said chain by means of the address of the first entry (414) of the chain in the Link table (420).
- * (806): the links associated with the path are registered in the Topology Database. The Topology Database inform The Routing Database each time:
 - a link on the path is no more operational,
 - the bandwidth reservation on a link of the path is modified.

Upon receipt of the registration request, the Topology Database sends these informations until the Routing Database notifies it to stop. The registration is cancelled when a link is deleted from the Routing Database. This mechanism allows to maintain an up-to-date Routing database. refer to Figure 12 for further details concerning the "Link Update" Procedure.

- * (807): the path characteristics (413) are updated with the information contained in the Topology Database. Then the procedure is complete and terminates in stop step (808).

Routing Database Update

The "Routing Database Update" process is a background task running periodically and preventively for preparing potential or alternate paths or for updating paths already stored in the Path Table (410). The process is executed in absence of connection request for not overloading the connection set up process. It comprises three procedures:

1. the "Alternate Path Computation",
2. the "Potential Path Computation", and
3. the "Path Recomputation".

When the Routing Database Update Process starts, the Alternate Path Computation Procedure is called first. When all paths stored in the Routing Database have been scanned for determining whether or not an alternate path must be computed for each of them, the Potential Path Computation Procedure is performed.

When all nodes stored in the Topology Database have been scanned for determining whether or not a potential path must be computed for each of them, the Path Recomputation Procedure is performed.

When all the paths stored in the Routing Database have been recomputed, the Alternate Path Computation Procedure is performed again. This is a never ending process which computes one path each time it is called.

Alternate Path Computation

If a path already stored in the Routing database is

- unique for a given destination node, an Alternate Path Computation is performed. This computation is done with the same path characteristics. However, the outgoing link from the origin node and the incoming link to the destination node are not selected for not using the same first and last links than the path already stored in the Routing Database. In case of a failure on one of those links, it will be then possible to reroute a lot of connections in a very short period of time.
- 5 Alternate paths are prepared in advance and are ready to be used.
- 10 Figure 9 shows a general flow chart of the Alternate Path Computation Procedure. let P be the current path to be scanned and NPR the next procedure to execute when the Routing Database Update process will be called again.
- 15 * (900): start of the procedure.
- 20 * (901): a test determines whether or not the path P is unique for its destination.
 - If the path is not unique, the procedure goes on with step (902).
 - (903): if the path is unique, an alternative path is computed and stored in the Routing Database using the Store Path Procedure detailed in Figure 8. Then, the process goes on with step (904).
- 25 * (904): a test determines whether or not there are more paths to scan in the Routine Database.
 - (905): if there are more paths to scan, P points to the next path stored in the Routing Database. The next path stored will be the current path to be scanned when the Routing Database update process will be called next time. Then the procedure is complete and terminates in stop step (907).
 - (906): if there is no more path to scan, P points to the first path stored in the Routing Database and NPR points to the Potential Path Computation Procedure. Then the process is complete and terminates in stop step (907).
- 30 * (902): a test determines whether or not there are more paths to scan in the Routing Database.
 - (908): if there are more paths to scan, P points to the next path stored in the Routing Database and the procedure goes on with step (901).
 - (909): If there is no more path to scan, P points to the first path stored in the Routing Database and NPR points to the Potential path Computation Procedure. The procedure goes on with step (910).
- 35 * (910): the Potential Path Computation Procedure
- 40
- 45
- 50
- 55

described in Figure 10 is performed.

Potential Path Computation

The Topology database is scanned for determining nodes which are not selected as destination node in any of the paths stored in the Routing Database. If such a node is found, a path is calculated with default parameters.

Figure 10 shows a general flow chart of the Potential Path Computation Procedure. Let N be the current node to be scanned and NPR the next procedure to execute when the Routing Database Update process will be called again.

- * (1000): start of the procedure.
- * (1001): a test determines whether or not the node N is a destination for, at least, one path stored in the routing Database.
 - If yes, the procedure goes on with step (1002).
 - (1003): if not, a potential path to N is computed. The new path is stored in the Routing Database using the Store Path Procedure as described in Figure 8. The procedure goes on which step (1004).
- * (1004): a test determines whether or not there are more nodes to scan in the Topology Database.
 - (1005): if there are more nodes to scan, N points to the next node stored in the Topology Database. The next node will be the current node to be scanned when the Routing Database update procedure will be called again. Then the process is complete and terminates in stop step (1007).
 - (1006): if there are no more nodes to scan, N points to the first node stored in the Topology Database and NPR to the Path recomputation Procedure. Then the process is complete and terminates in stop step (1007).
- * (1002): a test determines whether or not there are more nodes to scan in the Topology Database.
 - (1008): if there are more nodes to scan, N points to the next node stored in the Topology Database. Then, the procedure goes on with again step (1001).
 - (1009): if there are no more nodes to scan, N points to the first node stored in the Topology Database and NPR to the Path Recomputation Procedure. The procedure goes on with step (1010).
- * (1010): the Path Recomputation Procedure as described in Figure 11 is performed.

Path Recomputation

The paths stored in the Routing Database are recomputed with the path characteristics stored in the Path Table (410) (to determine whether or not they are yet the best ones). If another better path is found, it is added in the Routing Database. Figure 11 shows a general flow chart of the Path Recomputation Procedure. Let P be the current path to be scanned and NPR the next procedure to perform when the Routing Database Update Process will be called again.

- * (1100): start of the procedure.
- * (1101): the path P is recomputed with stored characteristics and default parameters. The new path is stored in the Routing Database using the Store Path Procedure as described in Figure 8.
- * (1102): a test determines whether or not there are more paths to scan in the Routing Database.
 - (1103): if there are more paths to scan, P is initialized to the next path stored in the Routing Database. The next path will be current path to be scanned when the Routing Database update procedure will be called again. Then the process is complete and terminates in stop step (1104).
 - (1105): if there is no more path to scan, P is initialized to the first path stored in the Routing database and NPR to the Alternate Path Computation Procedure. Then the procedure is complete and terminates in stop step (1104).

Link Update

Figure 12 shows a general flow chart of the Link Update Procedure. This procedure is called by the Topology Database when occurs a configuration update related to a link stored in the Routing Database.

- * (1200): start of the procedure.
- * (1201): a test determines whether or not the configuration update is related to a change in the link state (failure on a link).
 - If not, the procedure goes on with step (1202).
 - (1203): if yes, the registration of that failing link is cancelled in the Topology Database.
- * (1204): a test determines whether or not a link parallel to the failing link (a link connecting the same nodes), is available.
 - (1205): if no link is available, all link occurrences in the Link Table (420) and all paths using that link in the Path table (410) are deleted. The procedure is complete and terminates in stop step (1209).

- (1206): if a link is available, the failing link is replaced by the parallel one in all paths stored in the Path Table. The procedure goes on with step (1207).
- * (1207): the parallel link is registered in the Topology Database.
- * (1208): the paths are updated with the parallel link characteristics (423). Then the process is complete and terminates in step (1209).
- * (1202): a test determines whether or not the configuration update is related to a link bandwidth reservation change.
 - If not, the process is complete and terminates in step (1209).
 - (1210): if yes, all link occurrences - Allocated bandwidth (423) - and all paths using that link - Free Bandwidth (413) - are updated in the Routing Database. Then the process is complete and terminates in step (1209).

Conclusion

The path calculated at the time the connection is requested is recorded in a Routing Database and updated each time a modification occurs in the network. Furthermore, alternate paths for supporting non-disruptive path switch on failure or preemption, and new paths towards potential destination nodes can be calculated and stored when the connection set up process is idle. These last operations are executed in background with a low processing priority and in absence of connection request.

Claims

1. A method of determining an optimal path between an origin node and a destination node in a packet switching communication network comprising a plurality of nodes (201..208) interconnected with transmission links (209), said method involving the steps of :

for each connection request, selecting a pre-calculated path satisfying said connection request in the routing database,

if no pre-calculated path satisfying the connection request is already stored in the routing database, calculating a path satisfying said connection request and storing said path within the routing database,

re-calculating the paths stored in the routing database periodically or in response to at least one pre-determined event in the network.

2. The method according to claim 1 wherein said pre-determined events in the network include :
 - 5 routing requests,
 - traffic variations,
 - network configuration changes.
- 10 3. The method according to anyone of the preceding claims comprising the further step of :
 - calculating, for each pre-calculated path stored in the routing database, at least one alternate path with the same origin node and the same destination node.
- 20 4. The method according to claim 3 wherein, compared with a pre-calculated path, the alternate path is calculated with a different outgoing link from the origin node and a different incoming link to the destination node.
5. The method according to anyone of the preceding claims comprising the further step of :
 - 25 calculating paths from the origin node to all possible destination nodes.
6. The method according to anyone of the preceding claims wherein:
 - 30 a pre-calculated path is removed from the routing database after a predetermined period of time without being selected.
- 35 7. The method according to anyone of the preceding claims wherein the steps of:
 - 40 re-calculating the paths stored in the routing database in response to at least one pre-determined event in the network,
 - 45 calculating, for each pre-calculated path stored in the routing database, at least one alternate path with the same origin and destination nodes,
 - 50 calculating paths between the origin node and all possible destination nodes.
 - 55 removing pre-calculated paths,
 - are executed in absence of connection request.
8. The method according to anyone of the preceding claims wherein said routing database is attached to a topology database containing the network configuration and traffic characteristics and updated by means of network control messages, and comprises :

a path table with multiple entries, each entry representing a pre-calculated routing path between an origin and a destination node and comprising :

5

- an identification of the destination node,
- characteristics of the path,
- an identification of the first link on the path starting from the origin node.

10

a link table with multiple entries, each entry representing a link on a pre-calculated path stored in the path table and comprising :

15

- an identification of the link,
- characteristics of the link,
- an identification of the path the link belongs to,
- an identification of the next link along the path starting from the origin node.

20

9. A node (300) for carrying out the method according to anyone of the preceding claims.

10. A packet switching communication network comprising at least one node (300) according to claim 9.

30

35

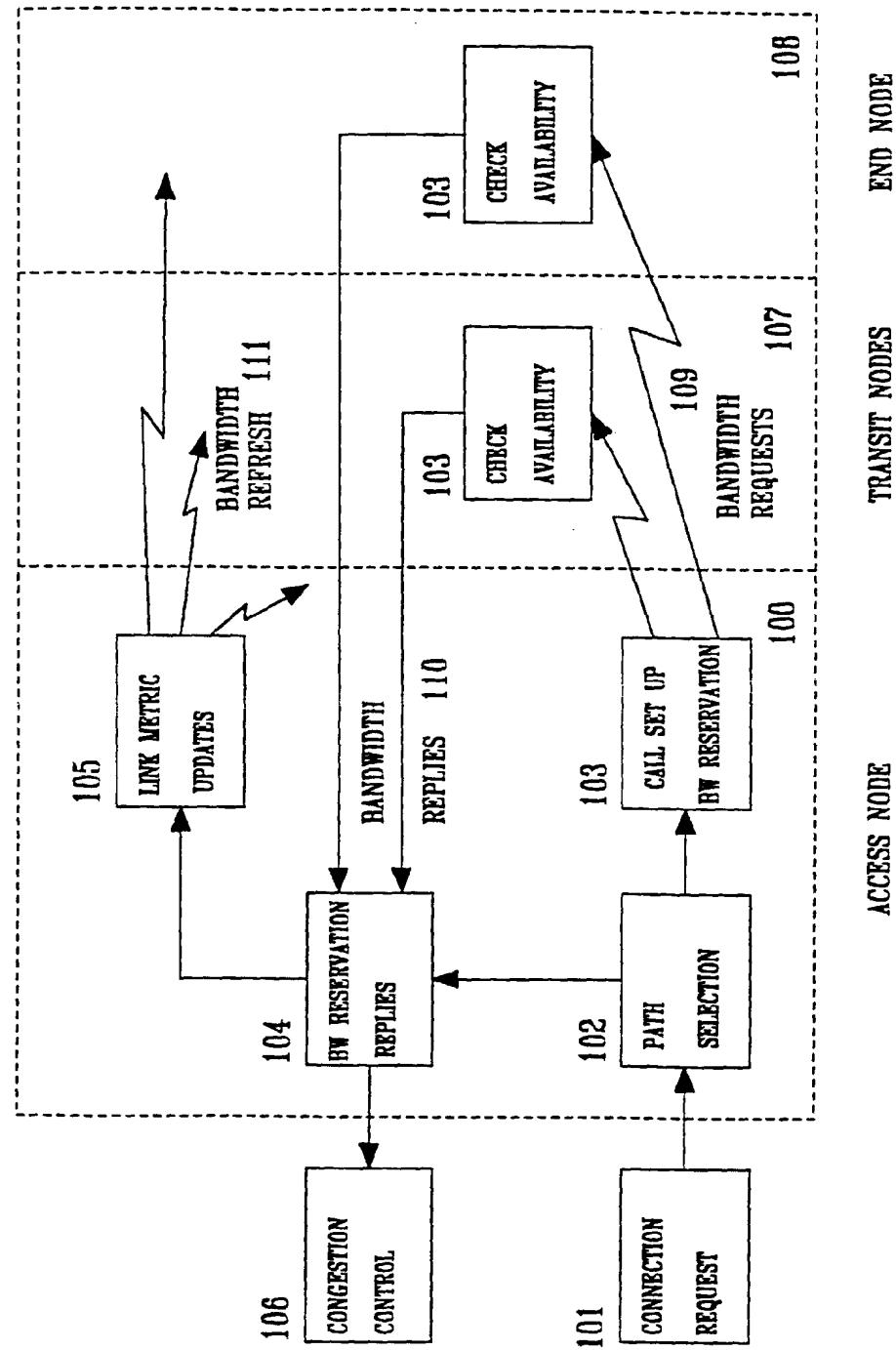
40

45

50

55

18

CALL SET UP
FIGURE 1

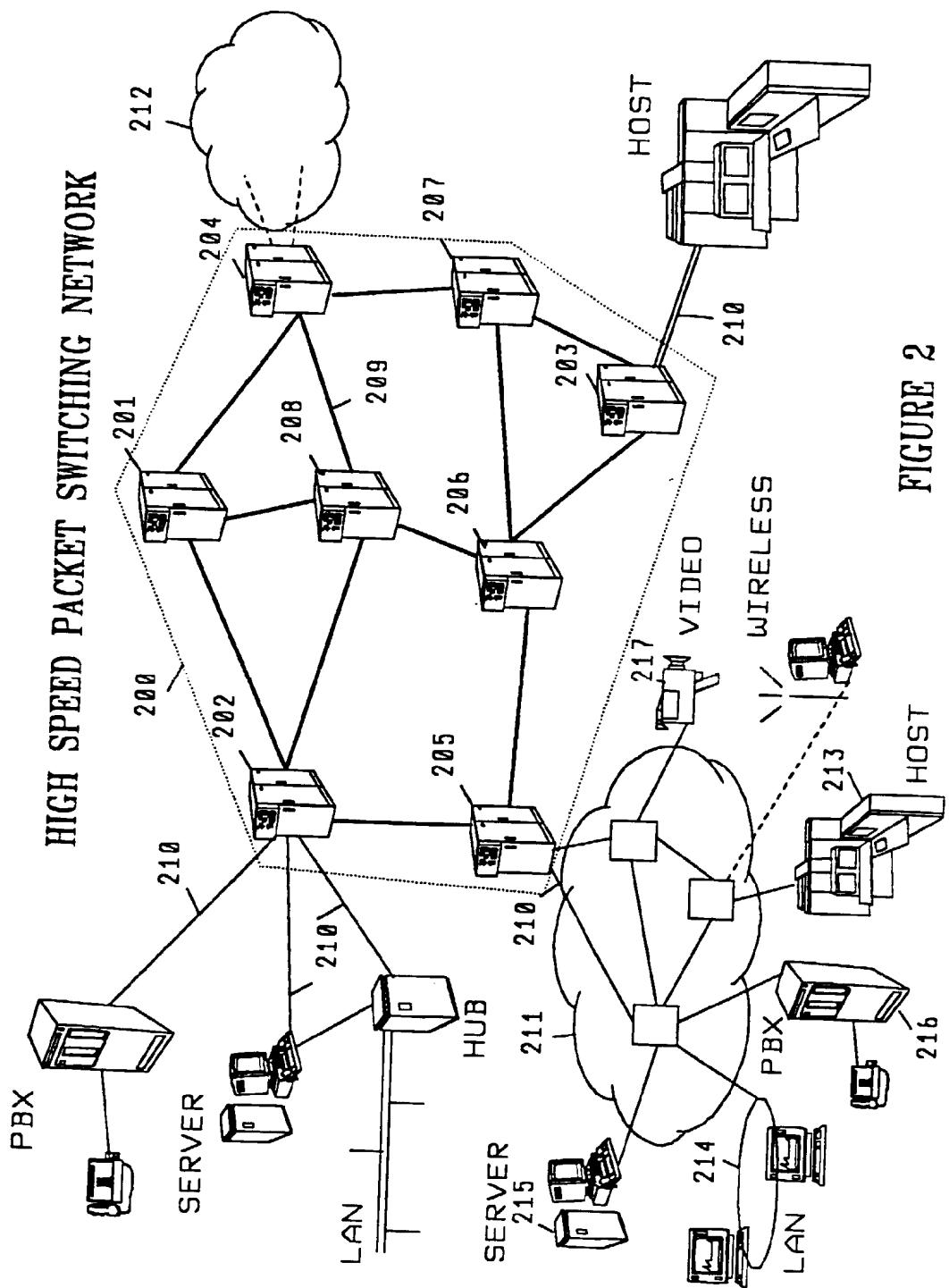


FIGURE 2

HIGH SPEED PACKET SWITCHING NODE

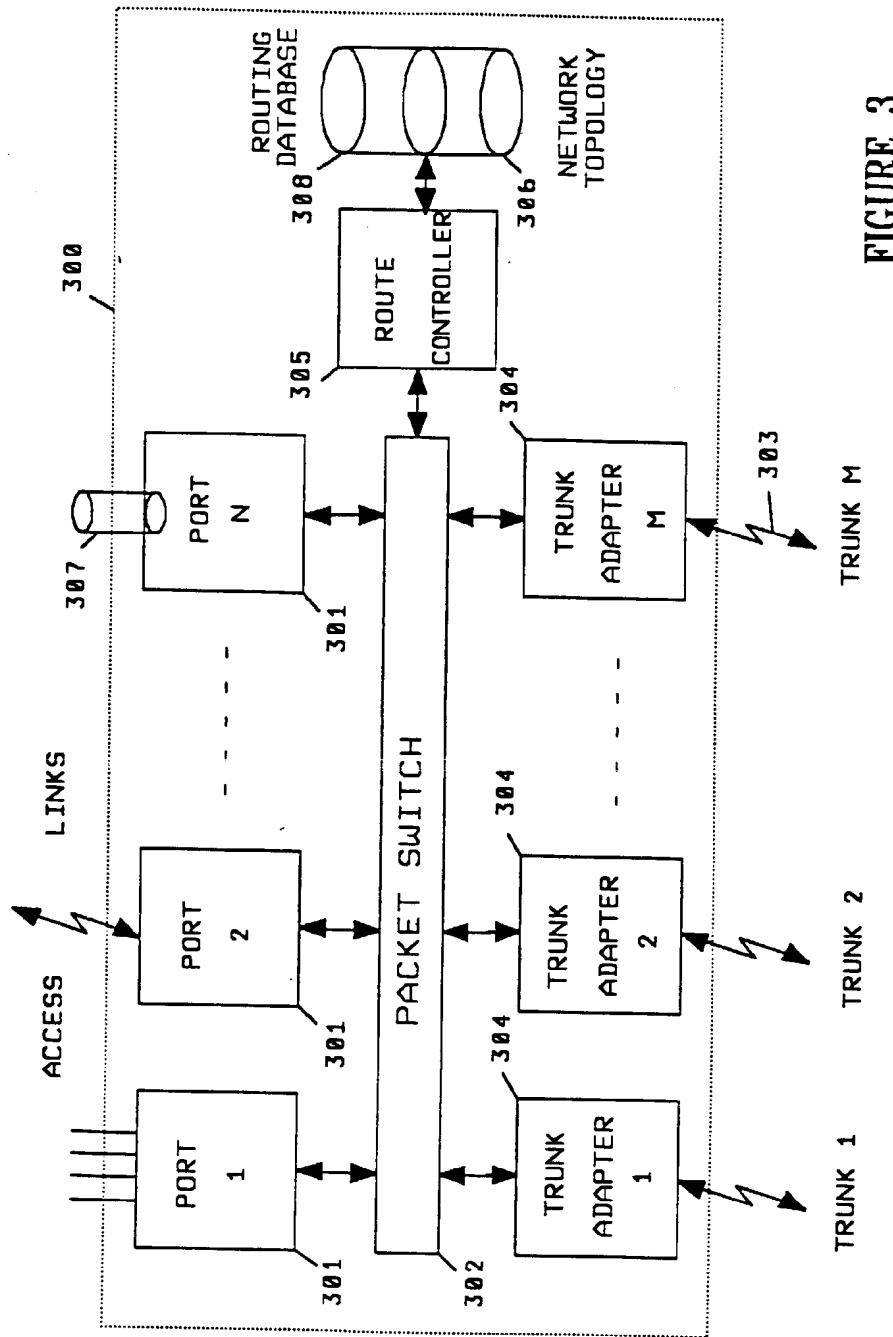


FIGURE 3

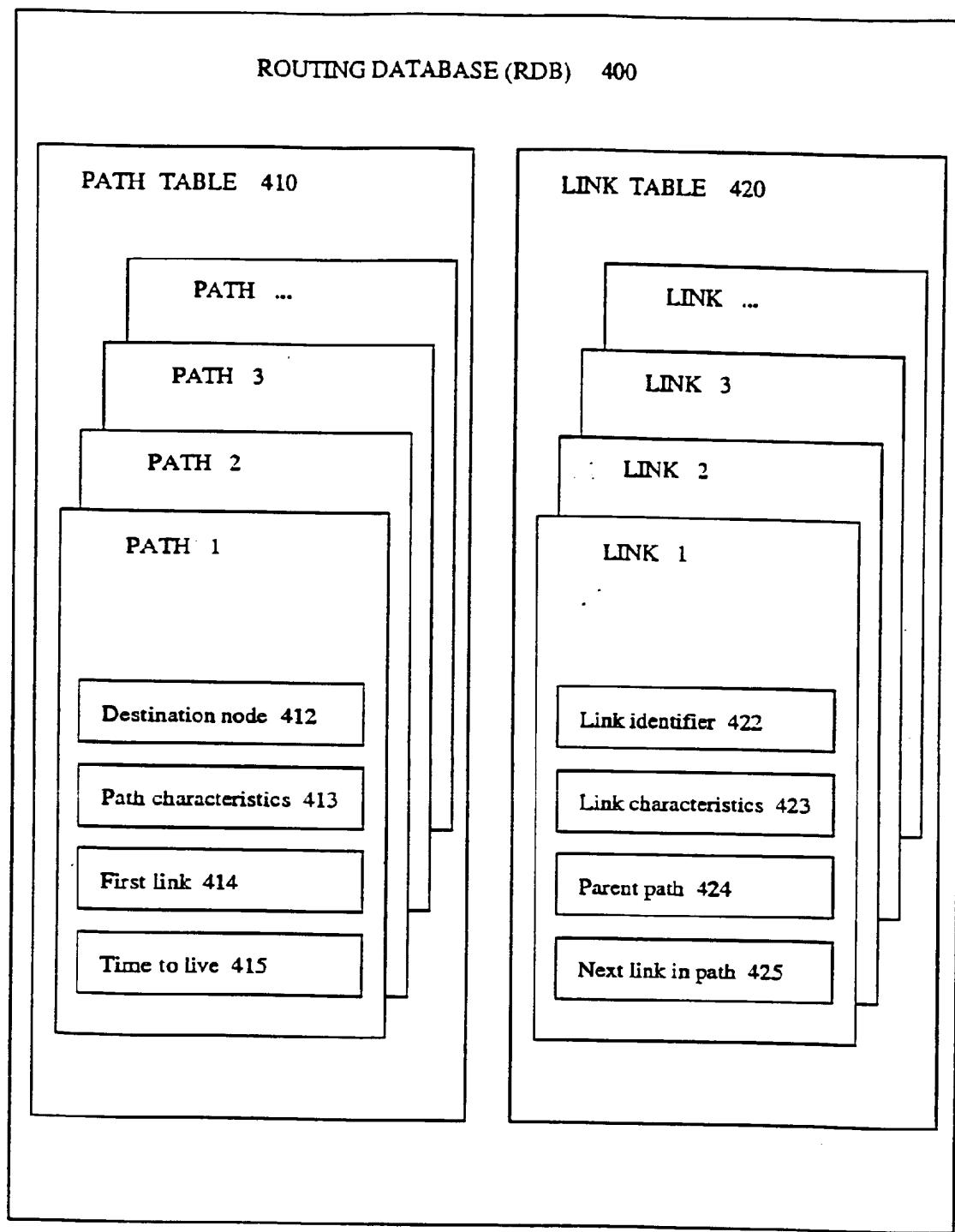
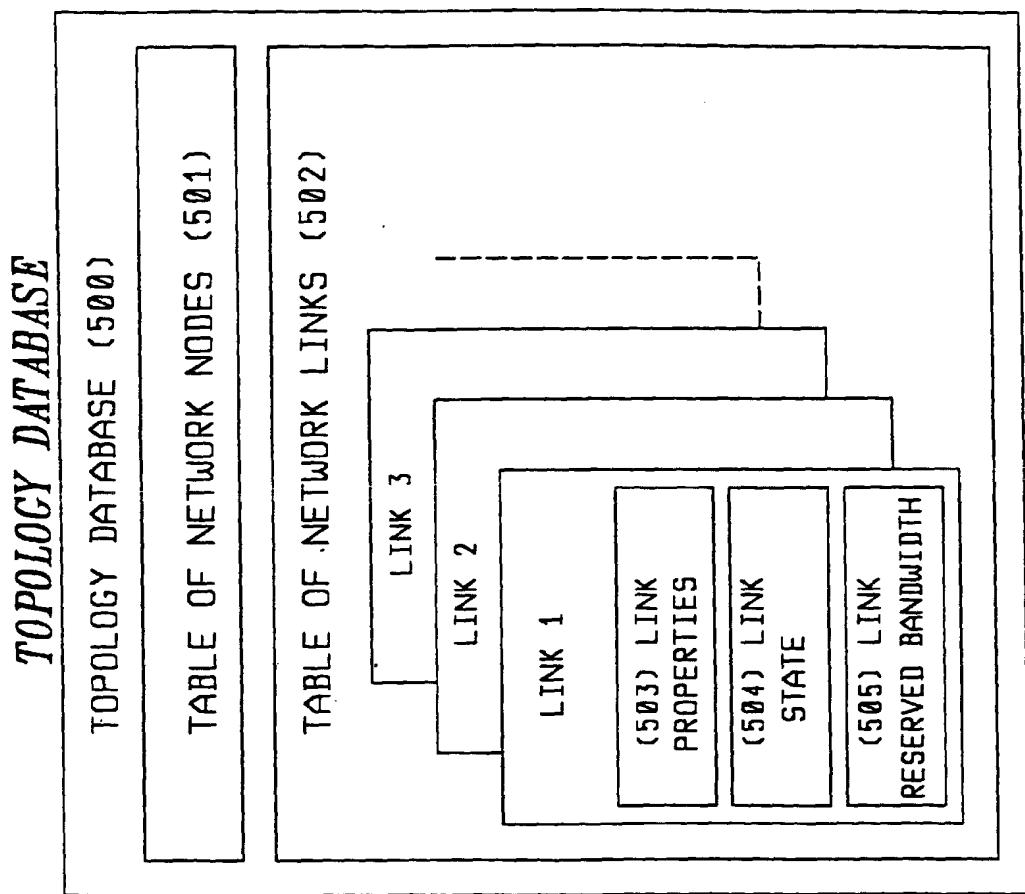


Figure 4 ROUTING DATABASE STRUCTURE

FIGURE 5



TOPOLOGY DATABASE

Link characteristics

<u>CHARACTERISTIC</u>		<u>LINK VALUES</u>			
<u>NAME</u>	<u>DESCRIPTION</u>	A	B	...	N
C	Total capacity (bps)	C_A	C_B	...	C_N
rf	Reservable fraction (%)	rf_A	rf_B	...	rf_N
\hat{C}_R	Total reserved bandwidth (bps)	$\hat{C}_{R,A}$	$\hat{C}_{R,B}$...	$\hat{C}_{R,N}$
M_{NR}	Total bandwidth used by NR traffic (bps)	$M_{NR,A}$	$M_{NR,B}$...	$M_{NR,N}$
mps	Maximum packet size (bytes)	mps_A	mps_B	...	mps_N

FIGURE 6

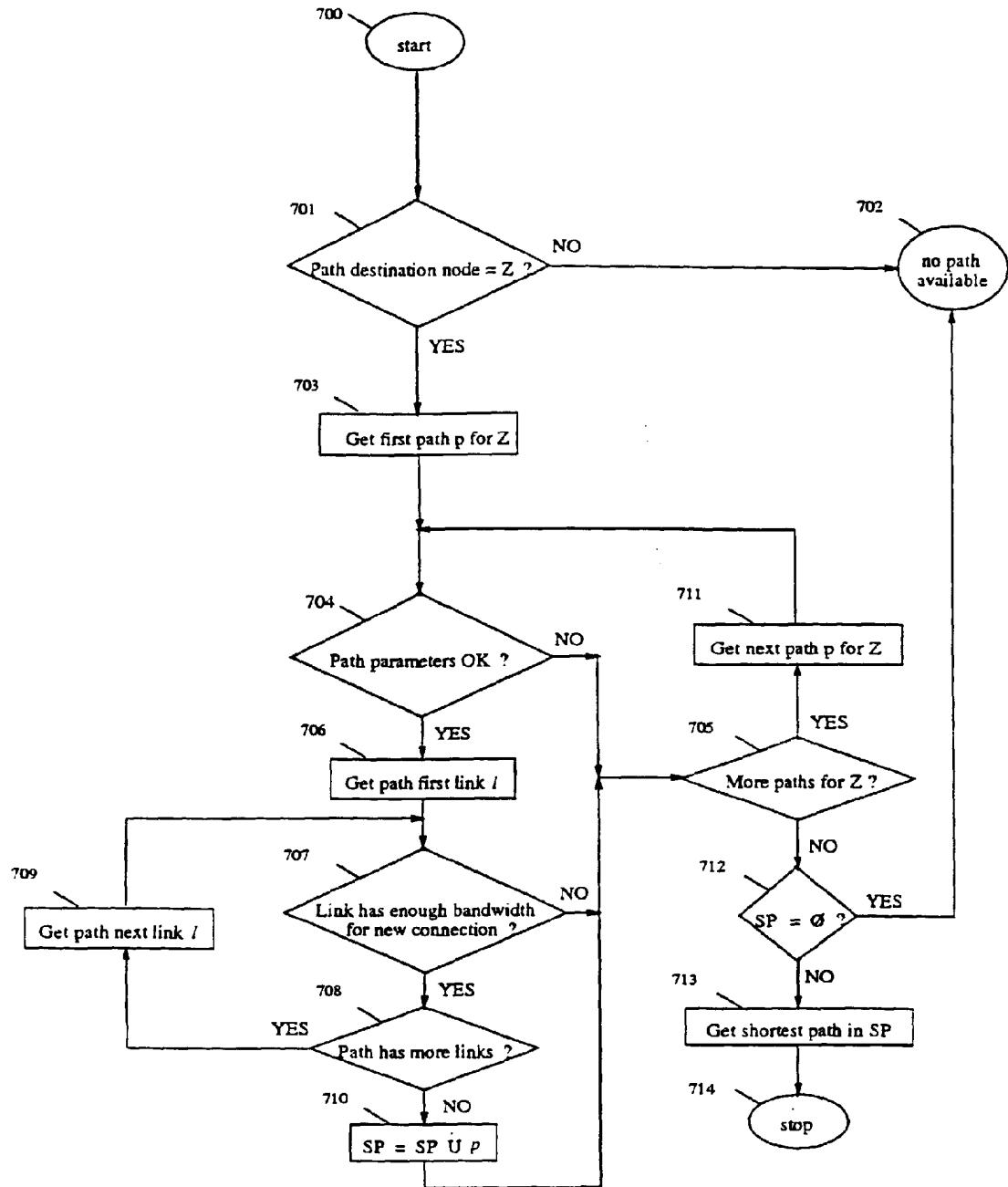


FIGURE 7 SEARCH PATH PROCEDURE

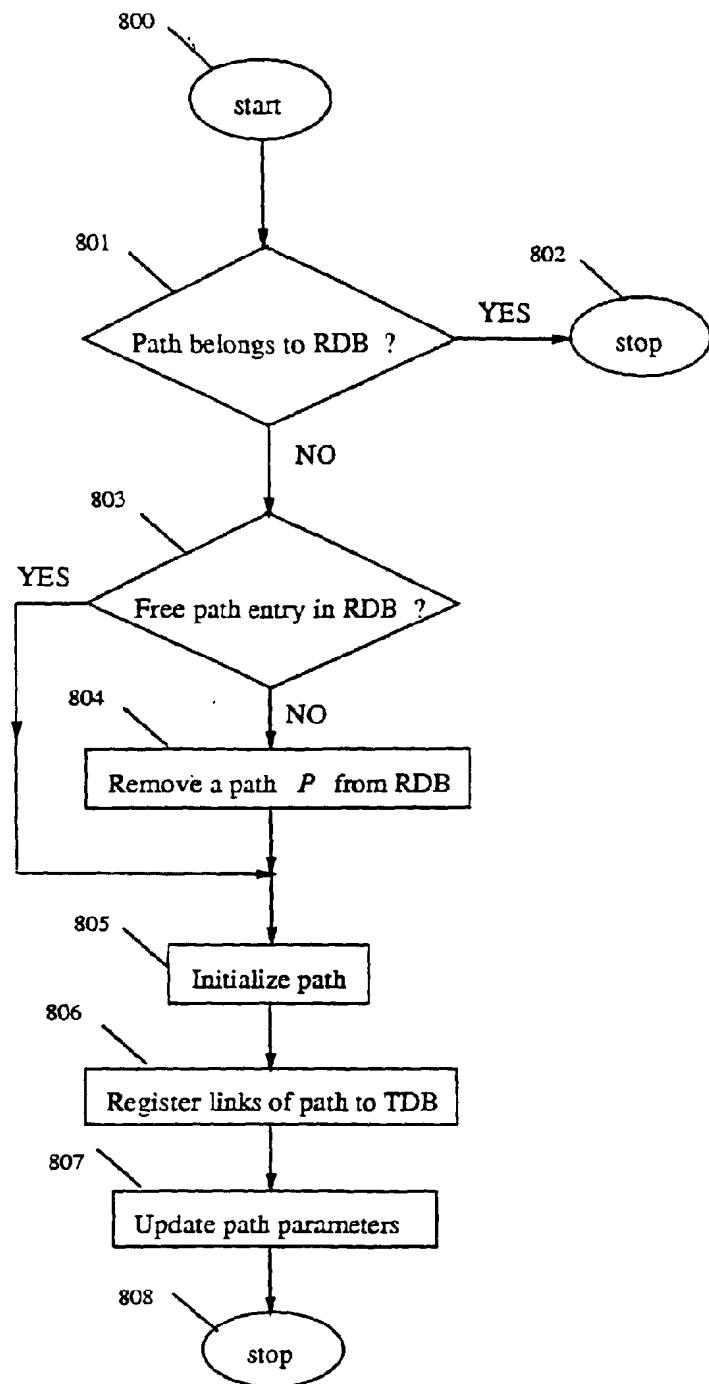


FIGURE 8 STORE PATH PROCEDURE

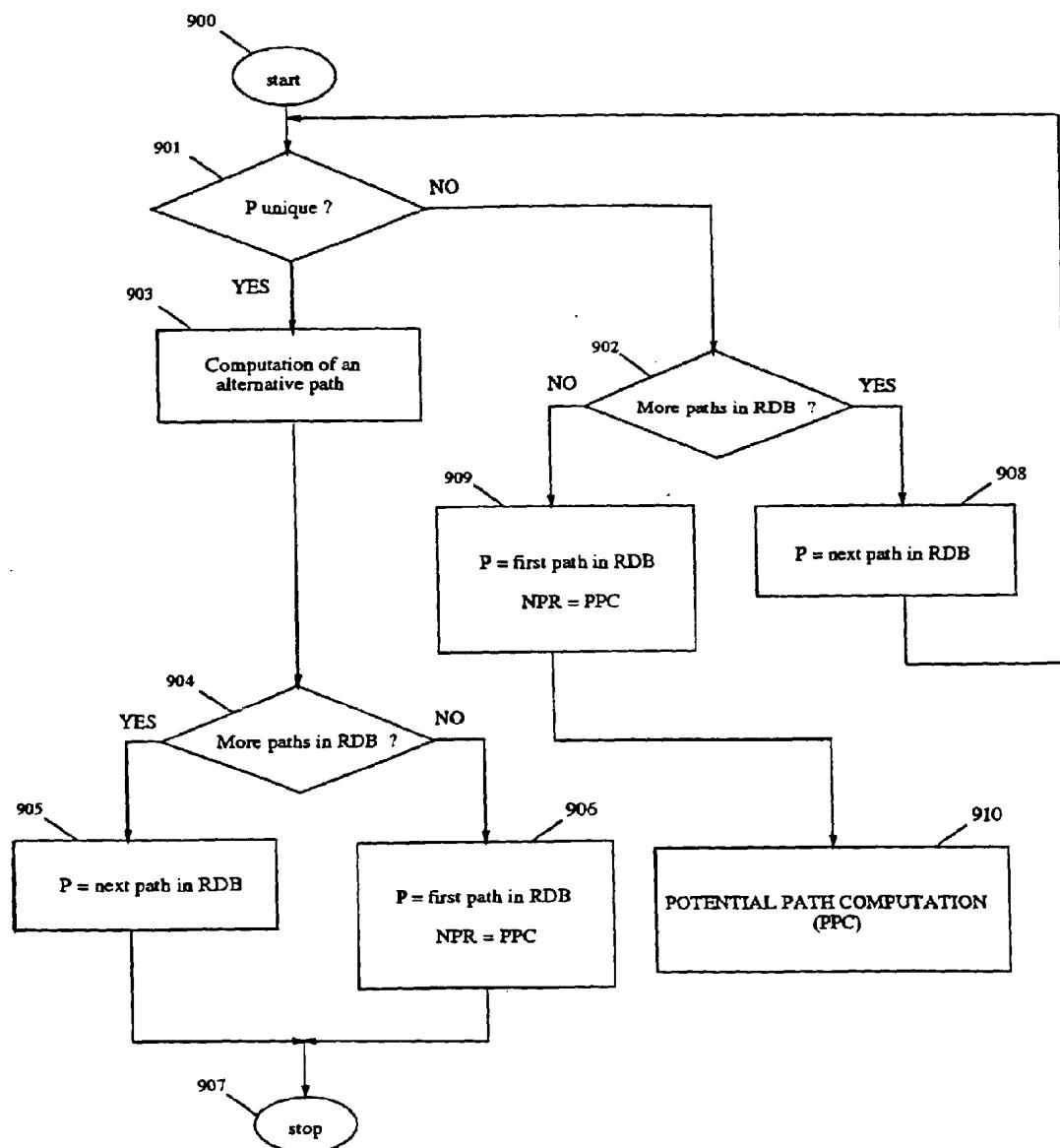


FIGURE 9 ALTERNATE PATH COMPUTATION PROCEDURE

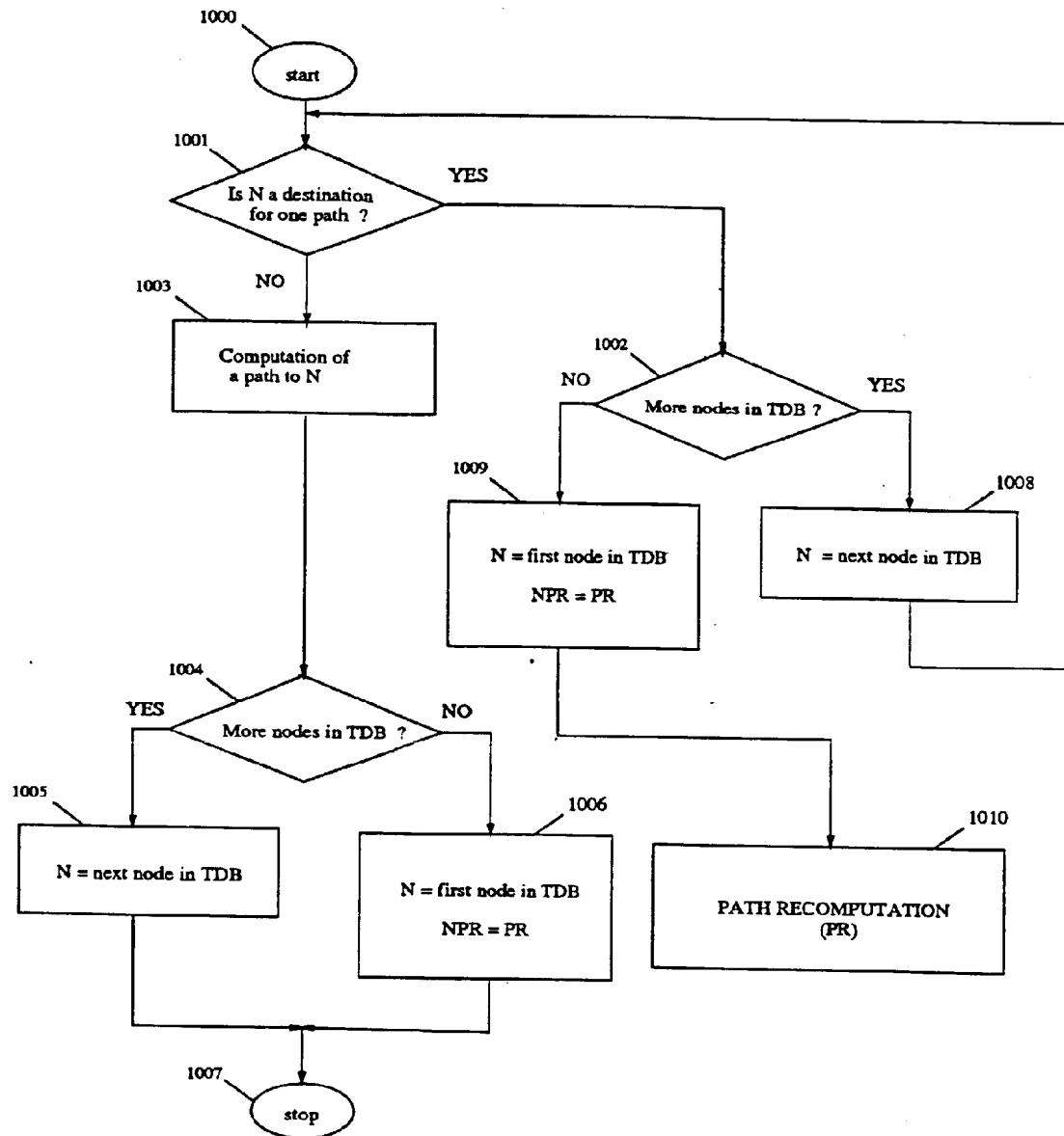


FIGURE 10 POTENTIAL PATH COMPUTATION PROCEDURE

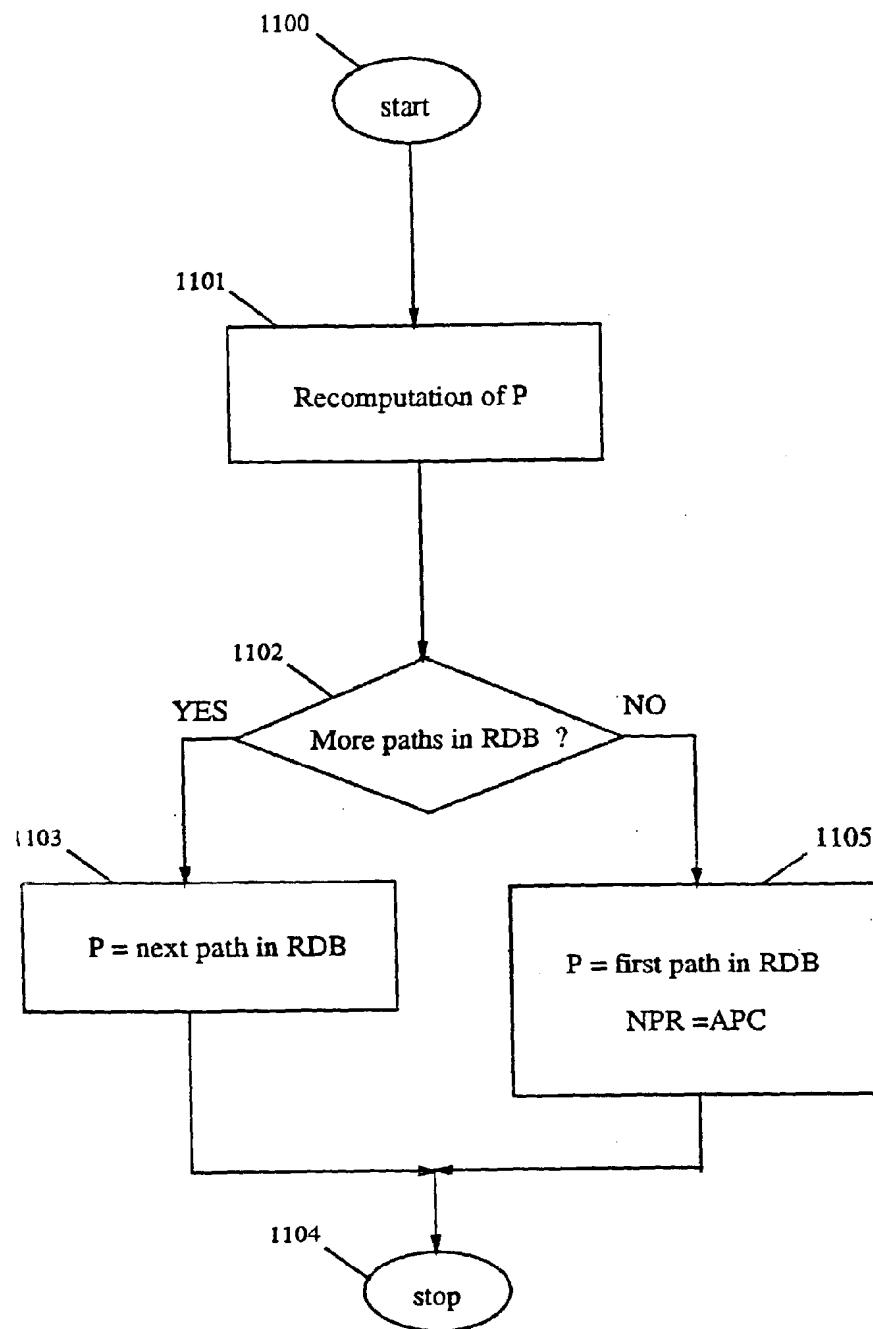


FIGURE 11 PATH RECOMPUTATION PROCEDURE

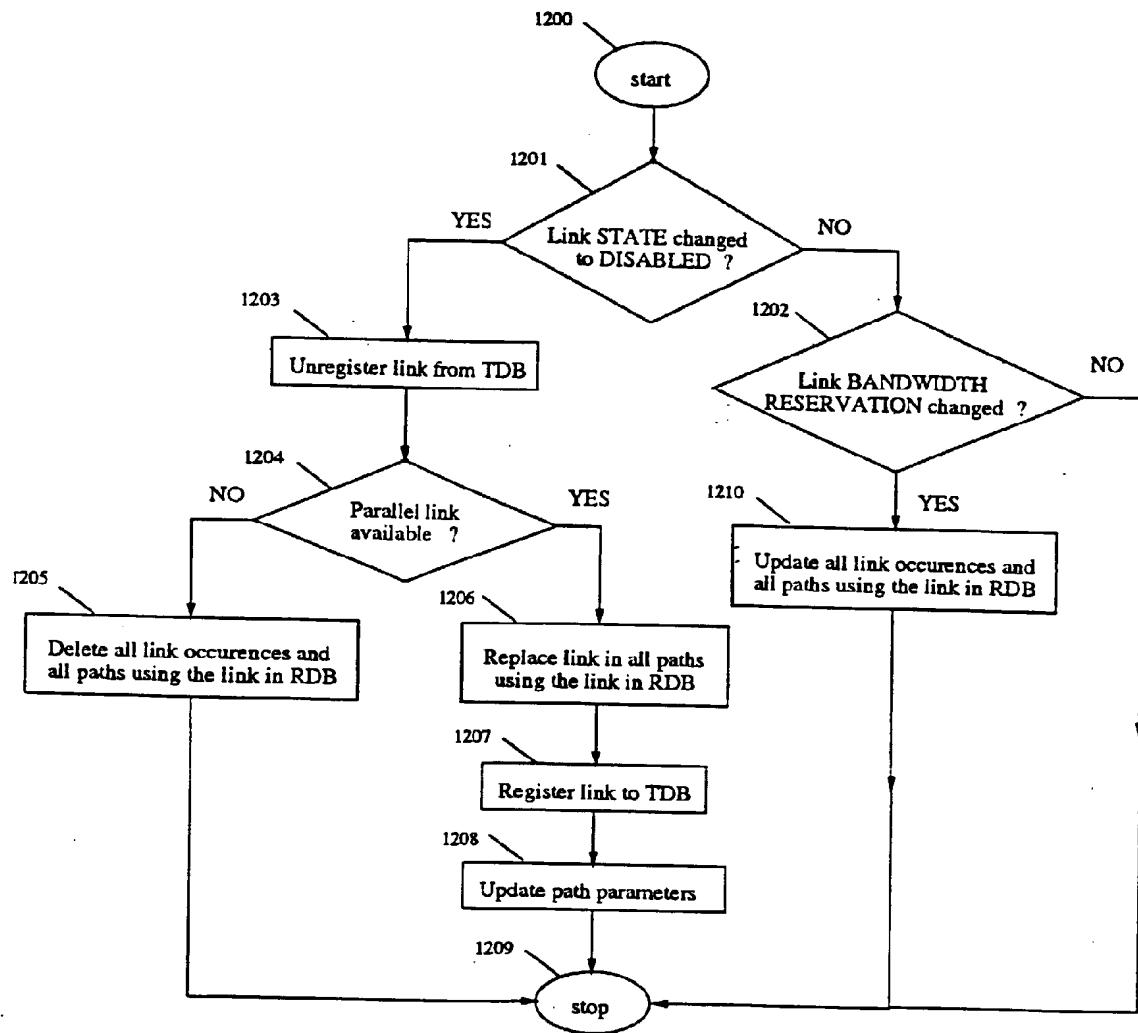


FIGURE 12 LINK UPDATE PROCEDURE

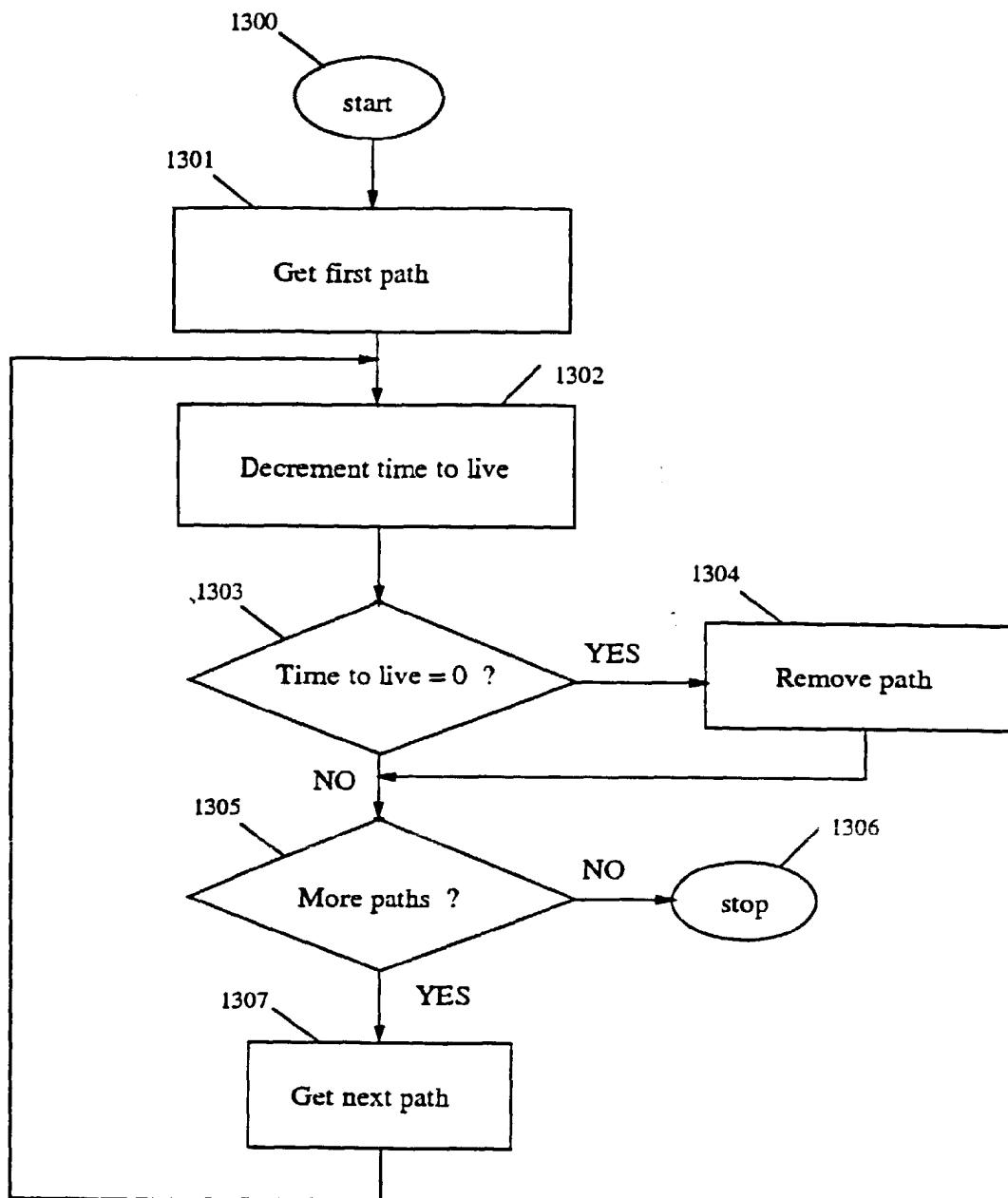


FIGURE 13 GARBAGE COLLECTION PROCEDURE